

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions of claims in the application:

Listing of Claims:

1. (Currently amended) A computer-implemented performance monitoring system, comprising:

a performance component that initiates at least one low-priority thread involving at least one computing resource and at least one high-frequency interrupt associated with at least one computing resource, the at least one low-priority thread comprising at least one selected from a group consisting of a memory-intensive operation thread [[and]] or a computationally-intensive operation thread, the at least one high-frequency interrupt comprising an interrupt with a frequency of at least approximately 300 Hertz, the at least one computing resource comprising at least one of a central processing unit (CPU) or a memory resource; and

a monitoring component that obtains at least one performance parameter for the at least one computing resource derived, at least in part, from at least one selected from the group consisting of the low-priority thread and the high-frequency interrupt initiated by the performance component, the monitoring component generates a report based on the at least one performance parameter upon the occurrence of a predetermined user-selected event, the at least one performance parameter comprising at least one of available CPU processing time, available memory, or available CPUs.

2. (Cancelled)

3. (Currently amended) The system of claim 1, the monitoring component tracks multiple CPU performance of at least one of a virtual CPU or a physical CPU~~the computing resource comprising at least one selected from the group consisting of a central processing unit (CPU) and a memory resource.~~

4. (Currently amended) The system of claim 1, the monitoring component is at least one of a dynamically linked library (DLL) associated with an operating system or embedded into the operating system~~the performance parameter comprising at least one from the group consisting of available central processing unit (CPU) processing time, available memory, and available CPUs.~~

5. (Cancelled)

6. (Currently amended) A computer-implemented method for facilitating computing system performance, comprising:

executing at least one low-priority thread involving at least one computing resource and at least one high-frequency interrupt related to the at least one computing resource, the at least one low-priority thread comprising at least one selected from a group consisting of a memory-intensive operation thread and a computationally-intensive operation thread, the at least one high-frequency interrupt comprising an interrupt with a frequency of at least approximately 300 Hertz, the at least one computing resource comprising at least one of a central processing unit (CPU) or a memory resource;

obtaining at least one performance parameter for the computing resource derived, at least in part, from at least one selected from the group consisting of execution of the low-priority thread and execution of the high-frequency interrupt, the at least one performance parameter comprising at least of available CPU processing time, available memory, or available CPUs; and

generating a report based on the at least one performance parameter upon the occurrence of a predetermined user-selected event.

7. (Cancelled)

8. (Currently amended) An entity employing the method of claim 6 comprising at least one selected from the group consisting of a light-weight operating system, a self-tuning application, a cell phone, a personal digital assistant (PDA), a logical ~~central processing unit (CPU)~~, an application programming interface (API), a computer, a server, and a handheld electronic device.

9. (Cancelled)

10. (Original) A device employing the system of claim 1 comprising at least one selected from the group consisting of a computer, a server, and a handheld electronic device.

11. (Currently amended) A computer-implemented performance monitoring system, comprising:

a performance component that initiates at least one low-priority thread related to at least one computing resource and at least one high-frequency interrupt involving at least one computing resource, the at least one low-priority thread comprising at least one of a memory-intensive operation thread or a computationally-intensive operation thread, the at least one high-frequency interrupt comprising an interrupt with a frequency of at least 300 Hertz, the at least one computing resource comprising at least one of a central processing unit (CPU) or a memory resource; and

a monitoring component that obtains at least one performance parameter for the computing resource derived, at least in part, from at least one selected from the group consisting of the high-frequency interrupt and the low-priority thread initiated by the performance component, the monitoring component generates a report based on the at least one performance parameter upon the occurrence of a predetermined user-selected event, the at least one performance parameter comprising at least one of available CPU processing time, available memory, or available CPUs.

12. (Cancelled)

13. (Currently amended) The system of claim 11, a low-priority thread is initiated on each CPU associated with the system to facilitate preventing an idle loop, the CPU comprising at least one of a logical CPU or a physical CPU~~the computing resource comprising at least one selected from the group consisting of a central processing unit (CPU) and a memory resource.~~

14. (Currently amended) The system of claim 11, the monitoring component interfaces with a plurality of physical CPUs, the plurality of physical CPUs are at least one of situated on a common die or at least a portion of the physical CPUs are situated on separate dies~~the~~

~~performance parameter comprising at least one from the group consisting of available central processing unit (CPU) processing time, available memory, and available CPUs.~~

15. (Cancelled)

16. (Currently amended) A computer-implemented method for facilitating computing system performance, comprising:

executing at least one high-frequency interrupt involving at least one computing resource and at least one low-priority thread related to at least one computing resource, the at least one high-frequency interrupt comprising an interrupt with a frequency of at least 300 Hertz, the at least one low-priority thread comprising at least one of a memory-intensive operation thread or a computationally-intensive operation thread, the computing resource comprising at least one of a central processing unit (CPU) or a memory resource;

obtaining at least one performance parameter for the computing resource derived, at least in part, from at least one selected from the group consisting of execution of the high-frequency interrupt and execution of the low-priority thread, the at least one performance parameter comprising at least one of available CPU processing time, available memory, or available CPUs; and

generating a report based on the at least one performance parameter upon the occurrence of a predetermined user-selected event.

17. (Cancelled)

18. (Currently amended) An entity employing the method of claim 16 comprising at least one selected from the group consisting of a light-weight operating system, a self-tuning application, a cell phone, a personal digital assistant (PDA), a logical ~~central processing unit (CPU)~~, an application programming interface (API), a computer, a server, and a handheld electronic device.

19. (Cancelled)

20. (Original) A device employing the system of claim 11 comprising at least one selected from the group consisting of a computer, a server, and a handheld electronic device.

21. (Previously presented) A computer-implemented performance monitoring system, comprising:

a performance component that initiates at least one low-priority thread involving at least one computing resource and at least one high-frequency interrupt involving at least one computing resource, wherein the low-priority thread comprising at least one selected from a group consisting of a memory-intensive operation thread and a computationally-intensive operation thread, the high-frequency interrupt comprising an interrupt with a frequency of at least approximately 300 Hertz, the computing resource comprising at least one selected from the group consisting of a central processing unit (CPU) and a memory resource, and the performance parameter comprising at least one from the group consisting of available central processing unit (CPU) processing time, available memory, and available CPUs; and

a monitoring component that obtains at least one performance parameter for the computing resource derived, at least in part, from at least one selected from the group consisting of the low-priority thread and the high-frequency interrupt initiated by the performance component, the monitoring component generates a report based on the at least one performance parameter upon the occurrence of a predetermined user-selected event.

22. (Cancelled)

23. (Cancelled)

24. (Cancelled)

25. (Cancelled)

26. (Currently amended) A computer-implemented method for facilitating computing system performance, comprising:

executing at least one low-priority thread involving at least one computing resource and at least one high-frequency interrupt involving at least one computing resource, wherein the at least one low-priority thread comprising at least one selected from a group consisting of a memory-intensive operation thread and a computationally-intensive operation thread, the at least one high-frequency interrupt comprising an interrupt with a frequency of at least approximately 300 Hertz, the computing resource comprising at least one selected from a group consisting of a central processing unit (CPU) and a memory resource;

obtaining at least one performance parameter for the computing resource derived, at least in part, from execution of at least one selected from the group consisting of the low-priority thread and the high-frequency interrupt, the at least one performance parameter comprising at least one selected from a group consisting of available CPU processing time, available memory, and available CPUs; and

generating a report based on the at least one performance parameter upon the occurrence of a predetermined user-selected event.

27. (Cancelled)

28. (Currently amended) An entity employing the method of claim 26 comprising at least one selected from the group consisting of a light-weight operating system, a self-tuning application, a cell phone, a personal digital assistant (PDA), a logical ~~central processing unit (CPU)~~, an application programming interface (API), a computer, a server, and a handheld electronic device.

29. (Cancelled)

30. (Original) A device employing the system of claim 21 comprising at least one selected from the group consisting of a computer, a server, and a handheld electronic device.

31. (Currently amended) A computer-implemented system that facilitates performance monitoring, comprising:

means for initiating at least one selected from a group consisting of at least one low-priority thread involving at least one computing resource and at least one high-frequency interrupt in at least one computing resource, wherein the at least one low-priority thread comprising at least one selected from the group consisting of a memory-intensive operation thread and a computationally-intensive operation thread, the at least one high-frequency interrupt comprising an interrupt with a frequency of at least approximately 300 Hertz, the computing resource comprising at least one selected from a group consisting of a central processing unit (CPU) and a memory resource;

means for obtaining at least one performance parameter for the computing resource derived, at least in part, from at least one selected from the group consisting of the low-priority thread and the high-frequency interrupt, the at least one performance parameter comprising at least one selected from a group consisting of available CPU processing time, available memory, and available CPUs; and

means for generating a report based on the at least one performance parameter upon the occurrence of a predetermined user-selected event.

32. (Cancelled)